

Change(s) applied³ At page 3, line ~~19~~, through page 4, line ~~4~~:
to document,

/N.B.H./

10/31/2011

DCOM has its roots in Microsoft's object technology, which has evolved over the last decade from DDE (Dynamic Data Exchange, a form of messaging between Windows programs), OLE (Object Linking and Embedding, embedding visual links between programs within an application), COM (the Component Object Model, used as the basis for all object binding), and ActiveX (COM enabled for the Internet) objects. As stated earlier, applications built from components are simply easier to debug and evolve than large, monolithic applications.

IN THE SPECIFICATION

Please amend the specification as follows:

8-17

Change(s) applied¹ At page 2, lines ~~9-19~~:
to document,

/N.B.H./

10/31/2011

One of the technologies underlying the World Wide Web is the prospect of using component software technology -- the idea of breaking large, complex software applications into a series of pre-built and easily developed, understood, and changed software modules called components -- as a means to deliver software solutions much more quickly and at a lower cost ~~(source: DCOM: A Business Overview, online at~~
~~<http://www.microsoft.com/ntserver/guide/dcom.asp>~~). The goal is to achieve economies of scale for software deployment across the industry.

Change(s) applied At page ~~19~~²:
to document,

/N.B.H/
10/31/2011

Fig 3 is a functional block diagram of a populated embodiment of the present invention showing a plurality of clients, adapters, and connectors. Client 0_L 92_L prepares service requests in a first format, such as ASP. Client 1_L 90_L prepares service requests in a second format, such as visual basic. Client m_L 94_L prepares service requests in a third format.

Each of the respective clients sends its service requests to the respective associated adapter as shown. Client 0_L 92_L service requests are sent to Adapter 0_L 84. Client 1_L 90_L service requests are sent to Adapter 1_L 86, and Client m_L 94_L service requests are sent to Adapter m_L 88. As explained above, each of these adapters (i.e., 84, 86, and 88) converts the inputs to a single format for input to GGATE 72.

The operation of GGATE 72 is explained in detail in the above cross-referenced U.S. Patent Applications. The converted and process service requests are transferred to the appropriate one of Connector 0_L 73, Connector 1_L 74, or Connector m_L 75. These connectors are arranged to handle the different formats as with the plurality of adapters. The converted, processed, and connected service requests are transferred to End Service Provider 78 for honoring.